

**IN THE DRAWINGS**

In the Office Action at page 4, item 8, the Examiner objected to the drawings. In order to overcome these objections, replacement figures are submitted herewith. The sheet containing FIG. 11 replaces the original sheet including FIG. 11, the sheet containing FIG. 12 replaces the original sheet including FIG. 12, the sheet containing FIG. 13 replaces the original sheet containing FIG. 13, and the sheet containing FIG. 18 replaces the original sheet containing FIG. 18. In FIGS. 11-13 and 18, the grayscale images (shaded areas) have been removed. Approval of these changes to the Drawings is respectfully requested.

For the convenience of the Examiner, an annotated sheet showing the changes made is attached. Approval of these changes to the Drawings is respectfully requested.



## REPLACEMENT SHEET

Fig. 11

```
//outline
// obtain line count of file
// return value
// line count of file
// explanation of parameter
// nothing
int getline()
{
    //1)Initialize variable
    int c,nl;
    nl=0;
    //2)count number of lines till EOF is detected
    while((c=getchar())!=EOF)
        //2.1)check whether it is line feed signal or not
        if (c=='\n')
            //2.1.1)count number of lines in the case of line feed signal
            ++nl;
    //3)display line count on screen
    printf ("%d\n", nl);
    //4)return line count to caller
    return nl;
}
```

Fig. 12

```
//outline
// obtain line count of file
// return value
// line count of file
// explanation of parameter
// nothing
int getline()
{
    //1)
    int c, nl;
    nl=0;
    //2)
    while ( (c=getchar()) !=EOF)
        //2.1)
        if (c=='\n')
            //2.1.1)
            ++nl;
    //3)
    printf ("%d\n", nl);
    //4)
    return nl;
}
```

REPLACEMENT SHEET





## REPLACEMENT SHEET

Fig. 13

```
//outline
// obtain line count of file
// return value
// line count of file
// explanation of parameter
// nothing
int getline()
{
    //1) initialize variable
    int c, nl;
    nl=0;
    //2) count number of lines till EOF is detected
    while ((c=getchar()) != EOF)
        //2.1)
        if (c=='\n')
            //2.1.1)
            ++nl;
    //3) display line count on screen
    printf ("%d\n", nl);
    //4)
    return nl;
}
```



## REPLACEMENT SHEET

Fig. 18

```
//outline
// obtain line count of file
// return value
// line count of file
// explanation of parameter
// nothing
int getline()
{
    //1) initialize variable
    int c, nl;
    nl=0;
    //2) count number of lines till EOF is detected
    while ((c=getchar()) != EOF)
        //2.1) check whether it is line feed signal or not
        if (c=='\n')
            //2.1.1) count number of lines in the case of line feed signal
            ++nl;
    //3) display line count on screen
    printf ("%d\n", nl);
    //4) return line count to caller
    return nl;
}
```